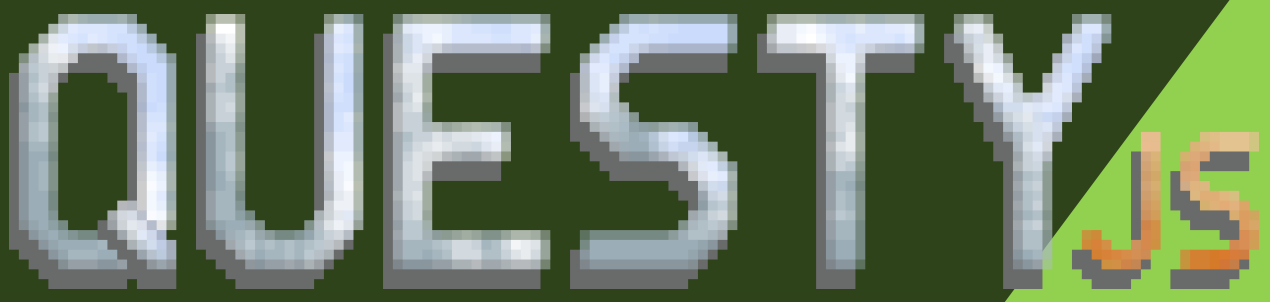


QyestyJs Prosjektrapport

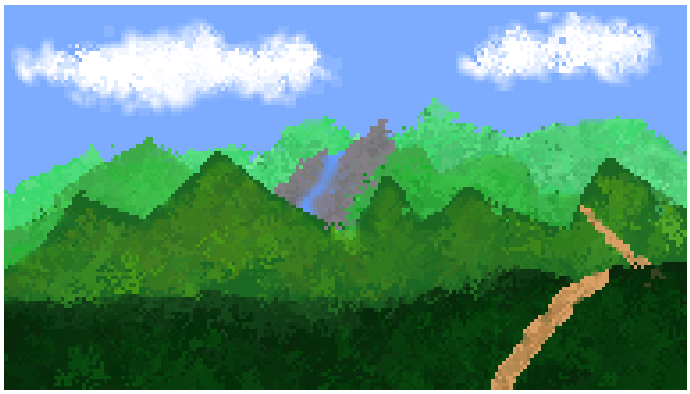
Martin Danielsen

Contents

ProsjektBesktivelse

Hva er QuestyJs

QuestyJs er et textbasert rollespill med grafiske elementer hvor målet ditt er å komme deg gjennom så mange rom som du klarer. Hvert rom har et tilfeldig trukket monster som du må beskjempe før du kan fortsette. Du registrerer deg inn med en bruker som blir lagret på en database. Siden brukeren din er lagret på en extern server så du kan fortstette der du stoppa sist gang hvor som helst.



Under the hood

Questyjs er webbasert og er skrevet i HTML, CSS, JavaScript og php. Den runner på en Apache server som har en Mariadb database. Denne databasen lagrer data om spillprogresjon, highscore, joindate, navn, brukernavn og passord. Alt av spilldata er lagret i arrays som optimaliserer filstørrelse og passordene i databasen er kryptert med Bcrypt som øker sikkerhet for brukere.

Prosjektplan

Tidsplan

Jeg lagde ikke en tidsplan til prosjektet mitt når jeg startet fordi jeg visste at jeg skulle bruke mange nye teknikker og tekologier som jeg måtte lære meg først. Dette gjorde at jeg ikke kunne forutse hvor lang tid noe ville ha tatt fordi jeg ikke visste hvor mye kunnskap jeg trengte.

Senere inn i prosjektet lagde jeg en tidsplan med kunnskapen om hvor realistisk jeg kunne følge den. Det er 2 punkter jeg ka forklare min tankegang bak. «Utvikle brukerstøtte» gikk gjennom nesten hele tidsplanen fordi da kunne jeg effektivt oppdatere dokumentasjon i mens jeg jobbet. «FAQ» er helt på slutten siden da ville jeg ha hatt mer innhold jeg kan bli spurt om.

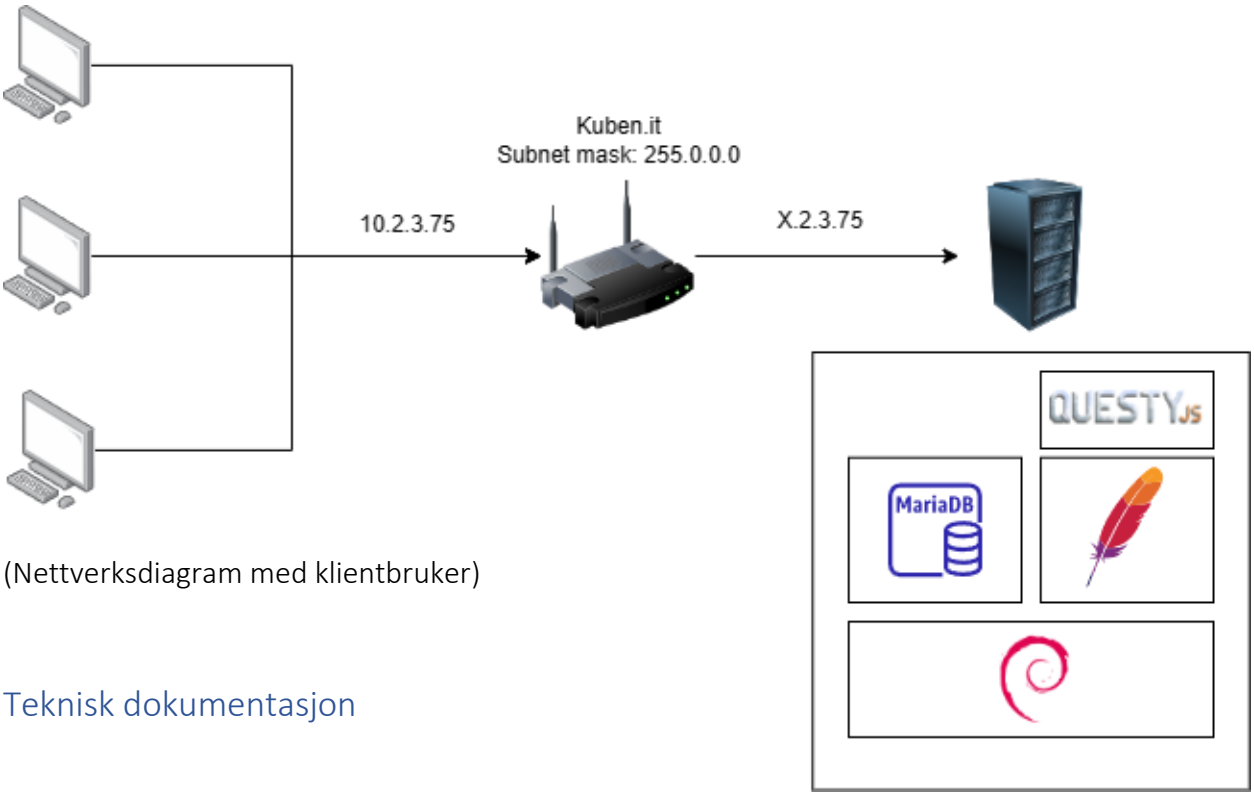
	Uke 1	Uke 2	Uke 3	Uke 4	Uke 5	Uke 6
Sette opp lamp server						
Utvikle brukerstøtte						
Assets						
FAQ						
Video						

Liste over oppgaver

Programmet jeg brukte til å bryte ned oppgaver til flere, enklere mål var «Google tasks». Google tasks lar deg lage oppgaver og underoppgaver. Dette gjør det enkelt å skrive ned alle målene dine og alle oppgaver du må gjøre for å fullføre de målene. Jeg ble mer ryddig og mer fokusert når jeg brukte denne metoden for planlegging.

Dokumentasjon

Nettverksdiagram



(Nettverksdiagram med klientbruker)

Teknisk dokumentasjon

Server pc (Backend)

- Ip adresse: 10.2.3.75
- Ssh: martin2imi@10.2.3.75
- Debian Login:
  - Brukernavn: martin2imi
  - Passord: MartinKub
- mariadb Login:
  - Brukernavn: root
  - Passord:

Klient pc

- Admin login
    - Brukernavn: Admin
    - Passord: 123

Github: https://github.com/kulbil/QuestyJs

Problemer og utfordringer

Xampp til debian server:

Når jeg skulle overføre spillet mitt og alt av dens backend ode til en helt ny infrastruktur oppsto det noen problemer. Jeg gikk fra ett enkelt xampp oppsett med apache og mysql til en egen satt opp debian11 server med apache og mariadb. Det oppsto 2 problemer:

Problem 1: Xampp kunne finne fram filer i mappene sine selv om filbanen man spesifiserte ikke vr helt riktig med store og små bokstaver. Dette ledet da til at når debian11 serveren skulle finne fram noen assets til spillet ble de ikke funnet på grunn av mine skrivefeil som gikk under radaren med xampp.

Problem 2: Koden klarte ikke å legge til nye rader og jeg brukte lang tid på å endre på premissions og finne andre mulige løsninger. Det viste seg at jeg brukte en gammel tabell constructor når jeg importerte den i den nye databasen, så «rank» kolonnen kunne ikke bli skrevet til fordi den ikke eksisterte.

Skrive om hele koden:

Når jeg først lagde QuestyJs het det bare Questy. Dette er fordi jeg skrev hele spillet i php før jeg skjønte konsekvensene av mine valg. Alle handlingene i spillet var utført med forms og get methods. Dette førte til at når jeg deploya spillet på en extern server ble ytelsen basert på nettverkfarten og prossesoren til serveren, som da ledet til at den ble ganske treig.

Jeg hadde med et uhell laget mitt egen lille sky baserte spill og lært meg hvorfor det som regel er en dårlig idé. Derfor skrev jeg om hele koden til spillet i javascript og lagde QuestyJs.

Kartlegging av lovverk

personopplysning(GDPR)

personopplysningloven er loven som beskytter sluttbrukerens personlige data som kan bli samlet. Dette kan for eksempel være navn, epost, lokasjon, osv. Denne loven passer på at dataen din ikke blir unødvendig samlet inn eller brukt uforsvarlig. Loven er relevant for mitt prosjekt fordi jeg samler inn navnet til brukeren, og kan senere implementere epost verifisering. Denne dataen er lagret på min database og må bli brukt ansvarlig.

Areidsmiljø loven

Arbeidsmiljøloven er loven som beskytter arbeidstakere. Dette betyr at du som arbeider har rett til en viss mengde betalte feriedager i året, maks timer man kan jobbe i en uke uten en pausedag, minstelønn, osv. Denne loven er ikke helt relevant for mitt prosjekt nå, men hvis jeg skulle vokse utviklingen og ansette folk, da må jeg passe på rettighetene deres.

Universiell utrofming

Universell utforming er en rekke krav som blir satt på utviklere som gjør de pliktig å lage deres programmer og nettsider tilgjengelig til mest mulig folk. Dette kan for eksempel være beskrivelser til bilder for blinde eller undertekst for svakhørte. Denne loven er relevant for meg fordi dette er en nettside som folk med potensiale handicapper kan bruke.

Markedsføringsloven

Markedsføringsloven er en lov som legger bestemte restriksjoner på reklamering og andre måter å tjene penger på. Dette kan for eksempel være restriksjoner på hva slags reklamer nettsiden din kan vise basert på hvem målgruppen din er. Barn kan ikke bli reklamert direkte til, så dette må u passe på. Det er også uloelig å reklamere for alkohol og tobakk. Dette er relevant for mitt prosjekt hvis jeg noen gans skal ha lyst til å legge til reklamer.

Åndverksloven

Åndsverkloven er loven som beskytter alle som skaper noe fra å få verket sitt stjålet/plagiert. Hvis du skriver en bok eller lager en tegning har du alle rettighetene til ditt verk og andre kan ikke kopiere deg uten din tillatelse. Alle bildene, monsterne og koden er laget av meg så jeg er beskyttet av denne loven.

Risikoanalyse

Mulig uønsket hendelse/ belastning	Vurdering av sannsyn-lighet	Vurdering av konsekvens:				Risiko-verdi
	(1-5)	Menneske (A-E)	Ytre miljø (A-E)	Øk/materiell (A-E)	Omdømme (A-E)	
Informasjon på datasen blir stjålet av en hacker	2	D	E	D	E	E
Spillet krasjer under bruk	4	A	B	D	C	C
Feilinformasjon gis til spilleren/sluttbruker	3	B	B	C	E	C
Spillet har en negativ påvirkning på spillerens mentale helse eller velvære	3	D	C	C	D	C
Spillet/nettsiden inneholder støtene eller diskriminerende innhold	1	D	C	B	D	B
Nettsiden virker ikke eller er veldig treig	5	A	A	D	B	D

Sannsynlighet vurderes etter følgende kriterier:

Svært liten 1	Liten 2	Middels 3	Stor 4	Svært stor 5
1 gang pr 50 år eller sjeldnere	1 gang pr 10 år eller sjeldnere	1 gang pr år eller sjeldnere	1 gang pr måned eller sjeldnere	Skjer ukentlig

Konsekvens vurderes etter følgende kriterier:

Gradering	Menneske	Ytre miljø Vann, jord og luft	Øk/materiell	Omdømme
E **Svært Alvorlig**	Død	Svært langvarig og ikke reversibel skade	Drifts- eller aktivitetsstans >1 år.	Troverdighet og respekt betydelig og varig svekket
D **Alvorlig**	Alvorlig personskade. Mulig uførhet.	Langvarig skade. Lang restitusjonstid	Driftsstans > ½ år Aktivitetsstans i opp til 1 år	Troverdighet og respekt betydelig svekket
C **Moderat**	Alvorlig personskade.	Mindre skade og lang restitusjonstid	Drifts- eller aktivitetsstans < 1 mnd	Troverdighet og respekt svekket
B **Liten**	Skade som krever medisinsk behandling	Mindre skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1uke	Negativ påvirkning på troverdighet og respekt
A **Svært liten**	Skade som krever førstehjelp	Ubetydelig skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1dag	Liten påvirkning på troverdighet og respekt

Egenvurdering

Hva har jeg lært?

Jeg har lært meg en rekke nye teknologier som nye kodespråk og driftløsninger. Jeg har også fått mye ærfaring bak planlegging og hvordan jeg skal jobbe med et prosjekt som har mange komponenter. Jeg har lært meg grunnleggende bruk av linux terminalen og filsystemet som da har gitt meg mer kunnskap om nettverk og hvordan datamaskiner komuniserer med hverandre ved hjelp av protokoller(IP, HTTP, TCP, osv).

Når det gjelder utvikling har jeg lært meg en rekke nye løsninger, kodespråk og biblioteker. Har er noen eksempler på hva jeg snakker om:

- **PHP (kodespråk):**
  Jeg brukte de første 2 månedene på å lære meg kodespråket php som er et språk som kjører på serveren. Dette gjør det mulig å lage koblinger mellom klienter og databaser, lage sikker innlogging meg kryptering, og å sikre databasen din med å fysisk seperere den fra klienter. Jeg bruker php til å lagre data fra nettlesern til serveren.
- **SQL (kodespråk)**:
  Jeg bruker SQL eller «Structured Query Language» til å håndtere sortering og søking I databasen min. SQL er et kraftig språk som tillater deg å lett håndtere store mengder data I en tabell med bruk av nøkkelord og enkel syntax. Jeg bruker SQL til å hente fram, legge til og oppdatere brukere og brukerdata.
- **Jquery (Bibliotek):**
  Jquery er et javascript bibliotek som forbedrer DOM manipulasjon med innebygde funksjoner som gjør det lettere å utføre generelle javascript oppgaver. Disse kan for eksempel være funksjoner som lager og sletter elementer, lettere event listeners og enklere css manipulasjon. Jeg bruker jquery til å kontrollere alt av det visuelle med spillet og å kjøre handlingene.
- **AJAX (KodeMetode):**
  AJAX (asyncronus javascript and xml) er en måte å kode på som gjør det mulig å hente data fra en server etter nettsiden har blitt lasted ned. Dette gjør det mulig å endre på hva brukeren ser uten å reloade hele siden. AJAX er ikke et kodespårk, men heller en måte du kan kode på ved hjelp av XMLHttprequest objekter og DOM. Jeg bruker AJAX til å oppdatere “Infovinduet” (referer til brukermanualen om hjelp) I spillet.

Vurdering

Hvis jeg skulle ha vurdert meg selv, min innsats og resultatet til arbeidet mitt så ville jeg da ha tatt i utganspunkt fra 3 temaer: Utvikling, drift, og brukerstøtte.

- Utvikling:
  Jeg har jobbet hardt med å utvikle et spill som ikke bare har en god brukeropplevelse, men en som også ser fint ut. Databasemetodene mine er sikkre og jeg har optimalisert koden så bra som en med min kunnskap kan. Noe jeg kan ha jobbet mer med er å planlegge litt bedre og forutse at problemer kommer til å oppstå.
- Drift:
  Gjennom denne oppgaven har jeg forsterket driftkunnskapen min mye. Jeg har satt opp en server med bare terminalen og jeg har forståelse bak alle protokoller som da gir meg mulighet til å mer effektivt feilsøke problemer. Noe jeg kan jobbe mer med er å forbedre kunnskapen min enda mer. Jeg kan lære meg flere løsninger til problemer som kan oppstå, og jeg kan forbedre sikkerheten i serverløsningen min.

Hva ville jeg ha gjort anderledes?

Hvis jeg skulle ha gjort noe anerledes så ville det ha vært å lage en google tasks liste mye tidligere i prossessen, begynne med javascript istedet for å starte om på alt, og lære meg serverløsningen min tidligere. Disse er ikke alt jeg ville ha gjort anderledes, men de gir innblikk på et hovedtema som da er å forbredre meg bedre før jeg begynner.

Kilder

W3schools:

PHP - https://www.w3schools.com/php/

AJAX - https://www.w3schools.com/js/js_ajax_intro.asp

Jquery - https://www.w3schools.com/jquery/

Stack overflow:

Hele siden - https://stackoverflow.com/

Digital ocean:

debian11 server - https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-debian-11

phpmyadmin - https://www.digitalocean.com/community/tutorials/how-to-install-phpmyadmin-from-source-debian-10